

For the final project in the course, we're tasked with creating/re-creating a 2D game in Unity, for my game I've selected Lunar Lander, the classic arcade game that is space-related... who'd have guessed it?!

So to start, let's deconstruct Lunar Lander.

**Overarching Goal:** To maneuver the lunar lander across the map and have it land safely on the planet's surface. Players can land anywhere, but specific locations offer a score multiplier. The challenge is to use the limited amount of fuel and time provided to adjust the lander's vertical and horizontal speed so it can come to a safe, smooth landing. Once the player lands, they go again until the timer runs out.

## Key Elements

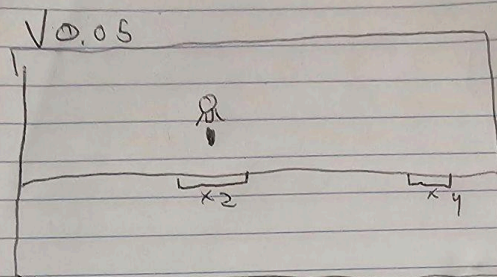
- **Gravity:** arguably the most important part of the game is gravity, as it's the primary force the player is fighting against, as it's what pulls the lander to the planet.
- **Lander:** The representation of the player, it'll have to follow the laws of gravity, and have a thruster on it that can apply force in the direction it's facing. As the thruster is on the bottom, the lander should be able to rotate and apply thrust in the direction it's going
- **Fuel & thrust:** Part of the Lunar Lander, the lander only has a set amount of fuel to use during a run, and it slowly drains as the thrusters are fired. The fuel also loses a random-ish amount when the lander crashes and respawns. This fuel amount will need to be carried over for each Run.
- **Map:** To start I'll leverage a strictly flat map, but Ideally the map will have hills and jagged edges that provide some variation and difficulty curves.
- **Score & Multiplier:** Provides a goal for the game and having various spots with multipliers encourages variation. Multipliers need to disappear after they are scored.
- **Timer:** As each run of the game is time-based, there will need to be a count-down timer that runs continuously. Once it hits 0, it'll display the score from the player's various attempts

Alright, I'm happy with this breakdown, now onto coding!

## Physical Notes

2 man tanker

Rough sketch



Want to start  
with a flat plain  
first!

I have gravity working, the Refinement is ~~not~~ needed  
but i need to get Rotation to work. So far I made  
it turn



Drawn like  
~~coin~~ coin



Over like a  
coin

but I can't get the  
rotation i want  
↓

Once I get rotation  
I can work on those  
direction

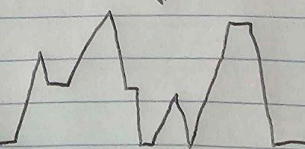
I'm going to try adjusting the axis' to see if that  
changes things.

Need to make more jagged map akin to ↓

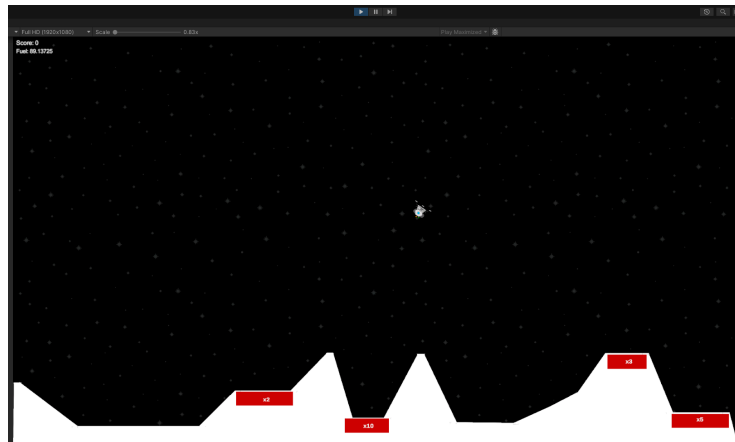
This will make it easier to denote  
tough to land multiplayer spots

~~to~~ Game tray marking it with  
boxes/triangles

While ~~amazing~~ at first the freeform sprite tool  
made this much easier



### In-Progress Screenshot



## Post-Mortem

This project more than any of the others I kept forgetting to do hand-written notes as it's not really my work style, I tend to think things through in my head, write them on a Google doc, or do a voice memo on my phone. Tho I did get a few hand-written notes done with sketches. So here's hoping between the typed, written, sketches, and code comments showing my thinking will work... tho It is something I'm going to push myself to work on more, as when I DO remember to write something down, I do find it at least a bit helpful.

On the actually coding side of the project, I've got a few key learnings:

- **OVER-COMMIT!**
  - Unlike normal life where you don't want to commit to too much, I've learned it's better to have too many commits than too little. While thankfully I didn't lose a bunch of work, I did have some instances where I had to re-adjust/re-write some things as I hadn't committed between making minor-ish changes. So gonna work on committing every time I tackle something or every 30ish minutes.
- **Check the branch**
  - After doing some initial work, I completely restarted the project to have a fresh slate and was making great progress. After a weekend working on it on my desktop, I went to work on it on my laptop and all my work was gone, I was left with the original version I scrapped... to say this caused stress was an understatement. Turned out a big part of the problem was that I had only updated my test branch and not the main, and was only pulling from main which had the outdated code.

- The other issue was I was using my OneDrive cloud folder as my git folder, which syncs on both computers, so it was cloud drives layered on top of version control... it caused some issues. So I'm working on transitioning my school folder from a constant one-drive to either a Google Drive or a private GitHub
- **Unity != Easy**
  - While I personally much preferred working in Unity over RayLib as the GUI was much appreciated, it does NOT make it easier. A perfect example is calculating the speed of the lander's fall. In RayLib I would have had to code how gravity worked, so I would have had a sense of how the object was affected by gravity.
  - On the flip side, I was lost in a sea of YouTube videos, unity manual pages, and Reddit threads full of users calling each other troglodytes for not knowing how to calculate velocity. I finally got a chance to ask Raph who explained it quickly and easily. Magnitude was the answer FYI.
- **Computers are not the same**
  - Even tho the game isn't all too taxing, it already operates differently on various machines. Despite having the same gravity settings on my desktop vs laptop, the game operated differently. I believe I solved the issue, but it's good learning to always playtest not only how a game runs on your rig, but on others too.

Overall I'm really happy with how the game turned out! There are some minor improvements I could make, such as the thruster PNG spinning after the first landing, or the rotation not being reset, but I know the more I look, the more I'll want to tweak and potentially break the game.

So in the end I loved the final project and learned that Unity is hard but I love it and if I ever have a coding question, it's better to go to Raph than the collective knowledge of humanity, because Raph is smarter and nicer.